REMARKS

These remarks are set forth in response to the office action mailed April 21, 2004 (the "Office Action"). As this amendment has been timely filed within the three-month statutory period, neither an extension of time nor a fee is required. Presently, claims 1-20 are pending in the Patent Application. Claims 1-20 have been rejected under 35 U.S.C. §103(a). The objections and rejections are set out in more detail below.

I Brief Review of Applicants' Invention

Prior to addressing the rejections on the art, a brief description of the present invention is appropriate. The present invention relates to a method and apparatus for high speed interprocess communications ("IPC"). In conventional IPC, multiple processes can communicate with one another via the use of a shared region of random access memory ("RAM") to which each process can write data, and from which each process can read data. When communicating through the shared region of RAM, a first process functioning as a message source can write a message to the shared region of RAM. Subsequently, a second process, functioning as a message receiver, can read the written message from the shared region of RAM. Thus, at a minimum, two system calls are required to move n bytes of data from the first process to the second process through the shared region of RAM. Moreover, 2*n bytes of data will be stored in total: n bytes into the shared region of RAM, and n bytes into user memory space associated with the second process.

To overcome the excessive overhead associated with conventional IPC utilizing a shared region of RAM, the high speed IPC method and apparatus of the present invention avoids moving 2*n bytes of data by passing to the second process a memory offset which can be used by the second process to access the data. Importantly, the second process can manipulate and modify the data in place within the shared region of RAM. Accordingly, it is not required that the data be copied from the shared region of RAM to an alternate location while the manipulation takes place, thereby improving system efficiency. In one arrangement, a locking mechanism can prevent access to the data from other processes while the data is being manipulated, and again grant access once the data manipulation is complete.

II. Claim Rejections on the Art

Presently, claims 1 through 20 are pending in the subject patent application (the "Application"). In the Office Action, however, each of claims 7 and 9 -12 have been rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,181,707 to Erickson et al. ("Erickson") in view of U.S. Patent No. 5,504,901 Peterson ("Peterson"). Claims 1-6, 8 and 13-18 have been rejected under 35 U.S.C. §103(a) as being unpatentable over Erickson in view of Peterson, and in further view of U.S. Patent No. 6,148,377 to Carter et al. ("Carter"). Finally, claims 19 and 20 were rejected under 35 U.S.C. §103(a) as being unpatentable over Erickson in view of Peterson, in view of Carter, and in further view of U.S. Patent No. 5,991,845 to Bohannon et al. ("Bohannon").

Referring to amended claims 1, 7 and 13, each of the amended claims recites the limitation of manipulating in the second process the accumulated data at the location in the message buffer where the data was accumulated from the first process. In consequence, system overhead that would otherwise be required to update the data is reduced. This limitation is disclosed on page 12, lines 19-25 of the specification.

As noted by the Examiner, such data manipulation is not taught or suggested by Erickson. The Examiner has, however, asserted that Peterson discloses the recited limitation at col. 7 lines 42-45 and col. 8 lines 62-65. Applicants' respectfully disagree with this assertion. The cited portions of Peterson's disclosure (with completed sentences) are as follows:

> This type of coded instructions is referred to as position independent code since the instructions do not depend upon loading the procedure to be called or the data to be accessed at memory addresses that are specified by the instructions. Instead, the data and procedures can be loaded into available memory space.
>
> Col. 7 lines 40-45

> More specifically, each memory location that stores an entry point address of a called procedure or the initial address of an accessed data set can be located relative to the memory

address of the instructions to call that procedure or access that data set by employing a memory offset pointer.

Col. 8 lines 60-65

Nowhere in either of these citations does Peterson teach or suggest manipulating in the second process the accumulated data at the location in the message buffer where the data was accumulated from the first process. Instead, the citation from col. 7 teaches that coded instructions do not depend upon loading a procedure at memory addresses specified by instructions, but that data and procedures can be loaded into available memory space. This citation does not disclose the recited limitation. There is no mention that the data and procedures are loaded into the available memory space by a first process, and then manipulated in place by a second process.

The citation from col. 8 teaches that the use of a memory offset pointer can be employed to locate an accessed data set. Again, this citation also fails to disclose the recited limitation. Importantly, there is to teaching or suggestion of manipulating the accessed data set at the location in the message buffer where the data was accumulated.

Notwithstanding the failure of the cited references to teach each of the limitations in amended claims 1, 7 and 13, Peterson is not analogous to Applicants' invention. Indeed, the present invention is directed to high speed interprocess communication achieved by sharing data at a memory location among multiple processes and manipulating the data in place. More particularly, the present invention avoids moving 2*n bytes of data by passing to the second process a memory offset which can be used by the second process to manipulate the data in place at the memory location identified by the memory offset.

Peterson is principally concerned with enabling a processing system to access data that are loaded into system memory at addresses that are not specified by the program code. Peterson's process, however, still requires consumption of a minimum of two memory copies, moving a total of 2n bytes of data in order to process n bytes. Thus, Peterson is not relevant to improving the speed of interprocess communications, but instead teaches a method of compiling program code without pre-determining memory addresses to be used by the program code.

III. Conclusion

Applicant has made every effort to present claims which distinguish over the prior art, and it is believed that all claims are in condition for allowance. Nevertheless, Applicant invites the Examiner to call the undersigned if it is believed that a telephonic interview would expedite the prosecution of the application to an allowance. In view of the foregoing remarks, Applicant respectfully requests reconsideration and prompt allowance of the pending claims.

Respectfully submitted,

Date: 7/21/04

Robert J. Sacco
Registration No. 35,667
Terry W. Forsythe
Registration No. 47,569
Sacco & Associates, P.A.
P.O. Box 30999
Palm Beach Gardens, FL 33420-0999
Tel: 561-626-2222